EECS 225B Problem Set 6

Due on 04/13/2020 at 9am on Gradescope

In this assignment, you explore applications of the projection slice theorem to tomography.

In this homework download, you should find `Pyramid.bmp`, a 256 pixels wide and 384 pixels tall image in 8-bit (256) gray. Using the `radon` function provided by Matlab, generate 180 equally spaced projections of the image. Take a moment to familiarize yourself with the capabilities of radon using the help files, and what it is computing for you. The radon function treats (192, 128) as the center of the image. For each projection, generate 465 sample points of $p_\theta(t)$ at equally spaced radius, roughly equivalent to one sample point per image unit.

Using the `iradon` function provided by Matlab, which implements the convolution (filtered) back-projection method, restore the original image. Save the reconstructed image as a bitmap file (`ConvBack.bmp`). Take a moment to familiarize yourself with the capabilities of `iradon` using the Matlab help files, and what it is computing for you. `iradon` includes an interpolation parameter (default linear), and a filter parameter (default Ram-Lak). How do these two parameters affect the convolution back-projection algorithm (refer to p. 429-432 in the textbook and lecture slides)? What happens if you reduce the number of θ samples by a factor of 2? By a factor of 5? Save these reconstructed images as well in bitmap files (`ConvBack2.bmp`, `ConvBack5.bmp`).

Implement the polar sampling method using nearest neighbor interpolation in the Fourier domain described in class, and compare it to the output of `iradon`. Save the reconstructed image as a bitmap (`Polar.bmp`).

Here are some helpful Matlab commands:

`X = fft2(x)`	Computes the 2D-DFT of a matrix x
`x = ifft2(X)`	Computes the inverse 2D-DFT of matrix X
`stuff = load('Phase.dat', 'mat')`	Loads the content of `Phase.dat` into the structure `stuff`.
`stuff.ImagePhase`	Access the `ImagePhase` matrix stored in `stuff`.

It might be helpful to look at the documentation for `radon` and `iradon`.
If you choose to do this homework in Python, you can use library functions analogous to the ones given above.

Note:

1. For each problem, you need to:
    1. Email your source code (zip it before you email) to eecs225bsp20@gmail.com if the question asks for any implementations.
        1. Make sure your code is executable. Either MATLAB or Python is okay. Please avoid C/C++ if possible (appreciate it!). If using Python, Jupyter Notebooks are preferred.
        2. Email title: FirstName_LastName_HW#. For example, Scott_McCrae_HW1.
        3. Submit a single PDF file on Gradescope which contains: i. your answer for each problem; ii. your source code (please also paste your source code here; screenshots are okay); iii. your output image(s). Make sure to prepare your solution to each problem on a separate page. On Gradescope, please select and match each page to the corresponding problems.
2. Remember to follow the course homework policies on collaboration. You're allowed to collaborate with other students on homework, but each student needs to write up their own original solutions. **Please do not submit the exact same writeup or code, as this is considered academic dishonesty. Copying solutions from any source is considered academic dishonesty.**